See the Assessment Guide for information on how to interpret this report.

ASSESSMENT SUMMARY

```
Compilation:  PASSED
API:          PASSED

SpotBugs:     FAILED (2 warnings)
PMD:          FAILED (6 warnings)
Checkstyle:   FAILED (0 errors, 28 warnings)

Correctness:  6/36 tests passed
Memory:       4/4 tests passed
Timing:       24/27 tests passed

Aggregate score: 47.78%
[ Compilation: 5%, API: 5%, Style: 0%, Correctness: 60%, Timing: 10%, Memory: 20% ]
```

ASSESSMENT DETAILS

```
The following files were submitted:
----------------------------------
 146 Jun  4 19:31 Outcast.java
3.5K Jun  4 19:31 SAP.java
4.2K Jun  4 19:31 WordNet.java


********************************************************************************
*  COMPILING
********************************************************************************


% javac SAP.java
*-----------------------------------------------------------

% javac WordNet.java
*-----------------------------------------------------------

% javac Outcast.java
*-----------------------------------------------------------


================================================================


Checking the APIs of your programs.
*-----------------------------------------------------------
SAP:

WordNet:

Outcast:

================================================================


********************************************************************************
*  CHECKING STYLE AND COMMON BUG PATTERNS
********************************************************************************


% spotbugs *.class
*-----------------------------------------------------------
M C MUI_USE_CONTAINSKEY MUI: Method WordNet.isNoun(String) calls keySet() just to call contains, use containsKey instead  At WordNet.java:[lin
M C MUI_USE_CONTAINSKEY MUI: Method WordNet.distance(String, String) calls keySet() just to call contains, use containsKey instead  At WordNet
SpotBugs ends with 2 warnings.


================================================================


% pmd .
*-----------------------------------------------------------
Outcast.java:2: Avoid unused parameter variables, such as 'wordnet'. [UnusedFormalParameter]
SAP.java:10: The private instance (or static) variable 'digraph' can be made 'final'; it is initialized only in the declaration or constructor
SAP.java:12: The private instance (or static) variable 'shortestCommonAn' can be made 'final'; it is initialized only in the declaration or co
WordNet.java:14: The private instance (or static) variable 'digraph' can be made 'final'; it is initialized only in the declaration or constru
WordNet.java:15: The private instance (or static) variable 'map' can be made 'final'; it is initialized only in the declaration or constructor
WordNet.java:16: The private instance (or static) variable 'storage' can be made 'final'; it is initialized only in the declaration or constru
PMD ends with 6 warnings.


================================================================


% checkstyle *.java
*-----------------------------------------------------------
[WARN] SAP.java:5:8: Unused import statement for 'java.util.List'. [UnusedImports]
[WARN] SAP.java:54:17: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:59:11: '//' or '/*' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:94:17: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:98:40: To specify an array type, put the square brackets before the variable name, e.g., 'String[] args' instead of 'String ar
[WARN] SAP.java:105:47: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:105:50: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:106:57: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:106:71: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] SAP.java:107:44: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:17:24: '//' or '/*' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:34:15: '//' or '/*' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:38:66: '//' or '/*' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:39:30: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:40:17: '}' is not followed by whitespace. [WhitespaceAround]
[WARN] WordNet.java:40:18: 'else' is not preceded with whitespace. [WhitespaceAround]
[WARN] WordNet.java:42:66: '//' or '/*' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:44:30: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:55:63: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:77:98: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:78:98: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:95:21: '}' is not followed by whitespace. [WhitespaceAround]
[WARN] WordNet.java:95:22: 'else' is not preceded with whitespace. [WhitespaceAround]
[WARN] WordNet.java:109:23: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:113:40: To specify an array type, put the square brackets before the variable name, e.g., 'String[] args' instead of 'Stri
[WARN] WordNet.java:114:46: ',' is not followed by whitespace. [WhitespaceAfter]
[WARN] WordNet.java:116:64: ',' is not followed by whitespace. [WhitespaceAfter]
Checkstyle ends with 0 errors and 27 warnings.

% custom checkstyle checks for SAP.java
*-----------------------------------------------------------
[WARN] SAP.java:1: In addition to the 5 required methods, you should define at least one private helper method to avoid code duplication. [Des
Checkstyle ends with 0 errors and 1 warning.

% custom checkstyle checks for WordNet.java
*-----------------------------------------------------------
[INFO] WordNet.java:1: The program uses neither 'DirectedCycle' nor 'Topological' to check whether the digraph is a DAG. [Design]

% custom checkstyle checks for Outcast.java
*-----------------------------------------------------------


================================================================


********************************************************************************
*  TESTING CORRECTNESS
********************************************************************************

Testing correctness of SAP
*-----------------------------------------------------------
Running 20 total tests.

Test 1: check length() and ancestor() on fixed digraphs
  * digraph1.txt
  * digraph2.txt
  * digraph3.txt

    java.lang.OutOfMemoryError: Java heap space

    java.base/java.util.LinkedList.linkLast(LinkedList.java:146)
    java.base/java.util.LinkedList.add(LinkedList.java:342)
    SAP.length(SAP.java:32)
    TestSAP.checkLengthAndAncestor(TestSAP.java:92)
    TestSAP.checkLengthAndAncestor(TestSAP.java:78)
    TestSAP.checkLengthAndAncestorFile(TestSAP.java:277)
    TestSAP.test1(TestSAP.java:428)
    TestSAP.main(TestSAP.java:1255)

  * digraph4.txt
  * digraph5.txt
  * digraph6.txt
  * digraph9.txt


WARNING: the time limit of 180 seconds was exceeded, so not all tests could be completed.
         This usually indicates a performance bug or an infinite loop.




Total: 0/20 tests passed:<font color = #990000><b> Could not complete tests, which results in a reported score of 0.</b></font>
================================================================
********************************************************************************
*  TESTING CORRECTNESS (substituting reference SAP)
********************************************************************************

Testing correctness of WordNet
*-----------------------------------------------------------
Running 14 total tests.

Test 1: check distance() with random noun pairs
  * 1000 pairs using synsets = synsets.txt; hypernyms = hypernyms.txt
    - nounA = Dendrolagus
    - nounB = Sacco
    - student   distance() = -2147483644
    - reference distance() = 13
    - failed on noun pair 1 of 1000
```

```
==> FAILED

Test 2: check distance() with all noun pairs
  * synsets = synsets15.txt; hypernyms = hypernyms15Path.txt
  * synsets = synsets15.txt; hypernyms = hypernyms15Tree.txt
  * synsets = synsets6.txt; hypernyms = hypernyms6TwoAncestors.txt
  * synsets = synsets11.txt; hypernyms = hypernyms11AmbiguousAncestor.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ModTree.txt
    - nounA = a
    - nounB = h
    - student   distance() = -2147483647
    - reference distance() = 4
    - failed on noun pair 7

  * synsets = synsets8.txt; hypernyms = hypernyms8WrongBFS.txt
  * synsets = synsets11.txt; hypernyms = hypernyms11ManyPathsOneAncestor.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ManyAncestors.txt
==> FAILED

Test 3: check distance() with random noun pairs
  * 1000 pairs using synsets = synsets100-subgraph.txt; hypernyms = hypernyms100-subgraph.txt
    - nounA = tetanus_immunoglobulin
    - nounB = jimhickey
    - student   distance() = -2147483645
    - reference distance() = 12
    - failed on noun pair 28 of 1000

  * 1000 pairs using synsets = synsets500-subgraph.txt; hypernyms = hypernyms500-subgraph.txt
    - nounA = blood_sugar
    - nounB = splinter
    - student   distance() = -2147483645
    - reference distance() = 13
    - failed on noun pair 7 of 1000

  * 1000 pairs using synsets = synsets1000-subgraph.txt; hypernyms = hypernyms1000-subgraph.txt
    - nounA = rennin
    - nounB = varicose_vein
    - student   distance() = -2147483648
    - reference distance() = 14
    - failed on noun pair 14 of 1000

==> FAILED

Test 4: check sap() with random noun pairs
  * 1000 pairs using synsets = synsets.txt; hypernyms = hypernyms.txt
    - student sap() is not a shortest common ancestor
    - nounA = waiver
    - nounB = solitudinarian
    - student   sap() = 'physical_entity'
    - reference sap() = 'entity'
    - student   distance() = -2147483644
    - reference distance() to 'entity' = 16
    - 'physical_entity' is not a common ancestor of 'waiver' and 'solitudinarian'
    - failed on noun pair 3 of 1000

==> FAILED

Test 5: check sap() with all noun pairs
  * synsets = synsets15.txt; hypernyms = hypernyms15Path.txt
  * synsets = synsets15.txt; hypernyms = hypernyms15Tree.txt
  * synsets = synsets6.txt; hypernyms = hypernyms6TwoAncestors.txt
  * synsets = synsets11.txt; hypernyms = hypernyms11AmbiguousAncestor.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ModTree.txt
    - student sap() is not a shortest common ancestor
    - nounA = a
    - nounB = h
    - student   sap() = 'd'
    - reference sap() = 'a'
    - student   distance() = -2147483647
    - reference distance() to 'a' = 4
    - 'd' is not a common ancestor of 'a' and 'h'
    - failed on noun pair 8 of 8

  * synsets = synsets8.txt; hypernyms = hypernyms8WrongBFS.txt
  * synsets = synsets11.txt; hypernyms = hypernyms11ManyPathsOneAncestor.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ManyAncestors.txt
==> FAILED

Test 6: check sap() with random noun pairs
  * 1000 pairs using synsets = synsets100-subgraph.txt; hypernyms = hypernyms100-subgraph.txt
    - student sap() is not a shortest common ancestor
    - nounA = immunoglobulin_G
    - nounB = jimhickey
    - student   sap() = 'protein'
    - reference sap() = 'entity'
    - student   distance() = -2147483645
    - reference distance() to 'entity' = 12
    - 'protein' is not a common ancestor of 'immunoglobulin_G' and 'jimhickey'
    - failed on noun pair 42 of 1000

  * 1000 pairs using synsets = synsets500-subgraph.txt; hypernyms = hypernyms500-subgraph.txt
    - student sap() is not a shortest common ancestor
    - nounA = acyl_group
    - nounB = fruit_sugar
    - student   sap() = 'monosaccharide monosaccharose simple_sugar'
    - reference sap() = 'unit building_block'
    - student   distance() = -2147483646
    - reference distance() to 'unit building_block' = 9
    - 'monosaccharide monosaccharose simple_sugar' is not a common ancestor of 'acyl_group' and 'fruit_sugar'
    - failed on noun pair 7 of 1000

  * 1000 pairs using synsets = synsets1000-subgraph.txt; hypernyms = hypernyms1000-subgraph.txt
    - student sap() is not a shortest common ancestor
    - nounA = stapes
    - nounB = immunoglobulin_D
    - student   sap() = 'protein'
    - reference sap() = 'thing'
    - student   distance() = -2147483646
    - reference distance() to 'thing' = 16
    - 'protein' is not a common ancestor of 'stapes' and 'immunoglobulin_D'
    - failed on noun pair 1 of 1000

==> FAILED

Test 7: check whether WordNet is immutable
  * synsets = synsets.txt; hypernyms = hypernyms.txt
==> passed

Test 8: check constructor when input is not a rooted DAG
  * synsets3.txt, hypernyms3InvalidTwoRoots.txt
    - constructor fails to throw an exception
    - it should throw a java.lang.IllegalArgumentException

  * synsets3.txt, hypernyms3InvalidCycle.txt
    - constructor fails to throw an exception
    - it should throw a java.lang.IllegalArgumentException

  * synsets6.txt, hypernyms6InvalidTwoRoots.txt
    - constructor fails to throw an exception
    - it should throw a java.lang.IllegalArgumentException

  * synsets6.txt, hypernyms6InvalidCycle.txt
    - constructor fails to throw an exception
    - it should throw a java.lang.IllegalArgumentException

  * synsets6.txt, hypernyms6InvalidCycle+Path.txt
    - constructor fails to throw an exception
    - it should throw a java.lang.IllegalArgumentException

==> FAILED

Test 9: check isNoun()
  * synsets = synsets.txt; hypernyms = hypernyms.txt
  * synsets = synsets15.txt; hypernyms = hypernyms15Path.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ModTree.txt
==> passed

Test 10: check nouns()
  * synsets = synsets.txt; hypernyms = hypernyms.txt
  * synsets = synsets15.txt; hypernyms = hypernyms15Path.txt
  * synsets = synsets8.txt; hypernyms = hypernyms8ModTree.txt
==> passed

Test 11: check whether two WordNet objects can be created at the same time
  * synsets1 = synsets15.txt; hypernyms1 = hypernyms15Tree.txt
    synsets2 = synsets15.txt; hypernyms2 = hypernyms15Path.txt
  * synsets1 = synsets.txt; hypernyms1 = hypernyms.txt
    synsets2 = synsets15.txt; hypernyms2 = hypernyms15Path.txt
==> passed

Test 12: call distance() and sap() with invalid arguments
  * synsets15.txt, hypernyms15Tree.txt, nounA = "x", nounB = "b"
  * synsets15.txt, hypernyms15Tree.txt, nounA = "b", nounB = "x"
  * synsets15.txt, hypernyms15Tree.txt, nounA = "x", nounB = "a"
  * synsets15.txt, hypernyms15Tree.txt, nounA = "x", nounB = "x"
  * synsets15.txt, hypernyms15Tree.txt, nounA = "a", nounB = null
  * synsets15.txt, hypernyms15Tree.txt, nounA = null, nounB = "a"
  * synsets15.txt, hypernyms15Tree.txt, nounA = null, nounB = null
  * synsets15.txt, hypernyms15Tree.txt, nounA = "x", nounB = null
  * synsets15.txt, hypernyms15Tree.txt, nounA = null, nounB = "x"
==> passed

Test 13: call isNoun() with a null argument
  * synsets15.txt, hypernyms15Path.txt
==> passed

Test 14: random calls to isNoun(), distance(), and sap(), with
         probabilities p1, p2, and p3, respectively
  * 100 random calls (p1 = 0.5, p2 = 0.5, p3 = 0.0)
    - failed on call 8 to distance()
    - nounA = absorbance
    - nounB = Alice_Malsenior_Walker
    - student   distance() = -2147483644
    - reference distance() = 14

  * 100 random calls (p1 = 0.5, p2 = 0.0, p3 = 0.5)
    - student sap() is not a shortest common ancestor
    - nounA = slime_bacteria
    - nounB = mythologisation
    - student   sap() = 'microorganism micro-organism'
    - reference sap() = 'entity'
    - student   distance() = -2147483646
    - reference distance() to 'entity' = 15
    - 'microorganism micro-organism' is not a common ancestor of 'slime_bacteria' and 'mythologisation'
    - failed on noun pair 3 of 100
    - failed on call 3 to sap()
  * 100 random calls (p1 = 0.0, p2 = 0.5, p3 = 0.5)
    - failed on call 2 to distance()
    - nounA = Pepin_III
    - nounB = source_book
    - student   distance() = -2147483647
    - reference distance() = 15

  * 100 random calls (p1 = 0.2, p2 = 0.4, p3 = 0.4)
    - failed on call 7 to distance()
    - nounA = confession_of_judgement
    - nounB = jury_mast
    - student   distance() = -2147483645
    - reference distance() = 18

==> FAILED


Total: 6/14 tests passed!


================================================================
********************************************************************************
*  TESTING CORRECTNESS (substituting reference SAP and WordNet)
********************************************************************************

Testing correctness of Outcast
*-----------------------------------------------------------
Running 2 total tests.

Test 1: check outcast() on WordNet digraph
        (synsets.txt and hypernyms.txt)
  * outcast2.txt
    - nouns = [Turing, von_Neumann]
    - student   outcast() = Hello
    - reference outcast() = Turing

  * outcast3.txt
    - nouns = [Turing, von_Neumann, Mickey_Mouse]
    - student   outcast() = Hello
    - reference outcast() = Mickey_Mouse

  * outcast4.txt
    - nouns = [probability, statistics, mathematics, physics]
    - student   outcast() = Hello
    - reference outcast() = probability

  * outcast5.txt
    - nouns = [horse, zebra, cat, bear, table]
    - student   outcast() = Hello
    - reference outcast() = table

  * outcast5a.txt
    - nouns = [earth, fire, air, water, heart]
    - student   outcast() = Hello
    - reference outcast() = heart

  * outcast7.txt
    - nouns = [Asia, Australia, North_America, India, Europe, Antarctica, South_America]
    - student   outcast() = Hello
    - reference outcast() = India

  * outcast8.txt
    - nouns = [water, soda, bed, orange_juice, milk, apple_juice, tea, coffee]
    - student   outcast() = Hello
    - reference outcast() = bed

  * outcast8a.txt
    - nouns = [Banti's_disease, hyperadrenalism, German_measles, gargoylism, Q_fever, amebiosis, anthrax, playboy]
    - student   outcast() = Hello
    - reference outcast() = playboy

  * outcast8b.txt
    - nouns = [apple, orange, banana, grape, strawberry, cabbage, mango, watermelon]
    - student   outcast() = Hello
    - reference outcast() = cabbage

  * outcast8c.txt
    - nouns = [car, auto, truck, plane, tree, train, vehicle, van]
    - student   outcast() = Hello
    - reference outcast() = tree

  * outcast9.txt
    - nouns = [lumber, wood, tree, leaf, nail, house, building, edifice, structure]
    - student   outcast() = Hello
    - reference outcast() = tree

  * outcast9a.txt
    - nouns = [hair, eyes, arm, mouth, nose, ear, cheek, brow, chin]
    - student   outcast() = Hello
    - reference outcast() = eyes

  * outcast10.txt
    - nouns = [cat, cheetah, dog, wolf, albatross, horse, zebra, lemur, orangutan, chimpanzee]
    - student   outcast() = Hello
    - reference outcast() = albatross
  * outcast10a.txt
    - nouns = [blue, green, yellow, brown, black, white, orange, violet, red, serendipity]
    - student   outcast() = Hello
    - reference outcast() = serendipity

  * outcast11.txt
    - nouns = [apple, pear, peach, banana, lime, lemon, blueberry, strawberry, mango, watermelon, potato]
    - student   outcast() = Hello
    - reference outcast() = potato

  * outcast12.txt
    - nouns = [Dylan, folk, Guthrie, idol, Minneapolis, music, musical, playing, public, recognition, review, thunderbird]
    - student   outcast() = Hello
    - reference outcast() = Minneapolis

  * outcast12a.txt
    - nouns = [competition, cup, event, fielding, football, level, practice, prestige, team, tournament, world, mongoose]
    - student   outcast() = Hello
    - reference outcast() = mongoose

  * outcast17.txt
    - nouns = [art, canvas, china, culture, kingdom, particularism, point, portable, ritual, road, script, sculpture, silk, style, transmissic
    - student   outcast() = Hello
    - reference outcast() = particularism

  * outcast20.txt
    - nouns = [art, Buddha, Buddhism, canvas, china, culture, India, kingdom, particularism, point, portable, ritual, road, script, sculpture,
    - student   outcast() = Hello
    - reference outcast() = particularism

  * outcast29.txt
    - nouns = [acorn, application, assembly, award, basic, cad, code, computer, custom, depth, development, finish, hardware, instruction, lar
    - student   outcast() = Hello
    - reference outcast() = acorn

==> FAILED

Test 2: check outcast() on WordNet subgraph
        (synsets50000-subgraph.txt and hypernyms50000-subgraph.txt)
  * outcast2.txt
    - nouns = [Turing, von_Neumann]
    - student   outcast() = Hello
    - reference outcast() = Turing

  * outcast3.txt
    - nouns = [Turing, von_Neumann, Mickey_Mouse]
    - student   outcast() = Hello
    - reference outcast() = Mickey_Mouse

  * outcast5.txt
    - nouns = [horse, zebra, cat, bear, table]
    - student   outcast() = Hello
    - reference outcast() = table

  * outcast5a.txt
    - nouns = [earth, fire, air, water, heart]
    - student   outcast() = Hello
    - reference outcast() = heart

  * outcast7.txt
    - nouns = [Asia, Australia, North_America, India, Europe, Antarctica, South_America]
    - student   outcast() = Hello
    - reference outcast() = India

  * outcast8.txt
    - nouns = [water, soda, bed, orange_juice, milk, apple_juice, tea, coffee]
    - student   outcast() = Hello
    - reference outcast() = bed

  * outcast8b.txt
    - nouns = [apple, orange, banana, grape, strawberry, cabbage, mango, watermelon]
    - student   outcast() = Hello
    - reference outcast() = cabbage

  * outcast8c.txt
    - nouns = [car, auto, truck, plane, tree, train, vehicle, van]
    - student   outcast() = Hello
    - reference outcast() = tree

  * outcast9.txt
    - nouns = [lumber, wood, tree, leaf, nail, house, building, edifice, structure]
    - student   outcast() = Hello
    - reference outcast() = tree

  * outcast10.txt
    - nouns = [cat, cheetah, dog, wolf, albatross, horse, zebra, lemur, orangutan, chimpanzee]
    - student   outcast() = Hello
    - reference outcast() = albatross

  * outcast11.txt
    - nouns = [apple, pear, peach, banana, lime, lemon, blueberry, strawberry, mango, watermelon, potato]
    - student   outcast() = Hello
    - reference outcast() = potato

==> FAILED


Total: 0/2 tests passed!
```

```
================================================================
********************************************************************************
*  MEMORY
********************************************************************************

Analyzing memory of SAP
*-----------------------------------------------------------
Running 1 total tests.

digraph G             = digraph-wordnet.txt
vertices in G         = 82192
edges    in G         = 84505
student     memory    = 8347944 bytes
reference   memory    = 10320584 bytes
ratio                 = 0.81
maximum allowed ratio = 2.50

Total: 1/1 tests passed!


================================================================



Analyzing memory of WordNet
*-----------------------------------------------------------
Running 3 total tests.

Test 1a: check memory of WordNet object
  * synsets = synsets1000-subgraph.txt; hypernyms = hypernyms1000-subgraph.txt
    - number of vertices in digraph = 1000
    - number of edges    in digraph = 1008
    - student   memory              = 927352 bytes
    - reference memory              = 1441648 bytes
    - student / reference ratio     = 0.6
    - maximum allowed rato          = 2.0

==> passed

Test 1b: check memory of WordNet object
  * synsets = synsets5000-subgraph.txt; hypernyms = hypernyms5000-subgraph.txt
    - number of vertices in digraph = 5000
    - number of edges    in digraph = 5059
    - student   memory              = 4520080 bytes
    - reference memory              = 7042912 bytes
    - student / reference ratio     = 0.6
    - maximum allowed rato          = 2.0

==> passed

Test 1c: check memory of WordNet object
  * synsets = synsets10000-subgraph.txt; hypernyms = hypernyms10000-subgraph.txt
    - number of vertices in digraph = 10000
    - number of edges    in digraph = 10087
    - student   memory              = 10859024 bytes
    - reference memory              = 16173480 bytes
    - student / reference ratio     = 0.7
    - maximum allowed rato          = 2.0

==> passed

Total: 3/3 tests passed!


================================================================



********************************************************************************
*  TIMING
********************************************************************************

Timing SAP
*-----------------------------------------------------------
Running 14 total tests.

Test 1: time SAP constructor
   *  digraph-wordnet.txt
      -  student solution time =  0.00 seconds
      -  maximum allowed  time =  1.00 seconds
==> passed

Test 2a-c: time length() and ancestor() with random pairs of vertices
  *  digraph-wordnet.txt
     -  reference solution calls per second:  768784.00
     -  student   solution calls per second:    4386.00
     -  reference / student ratio:               175.28

=> passed      student <= 50000x reference
=> passed      student <= 10000x reference
=> passed      student <=  5000x reference
=> passed      student <=  1000x reference

Test 3a-c: time length() and ancestor() with random subsets of 5 vertices
  *  digraph-wordnet.txt
     -  reference solution calls per second:  218386.00
     -  student   solution calls per second: 1872173.00
     -  reference / student ratio:                 0.12

=> passed      student <= 10000x reference
=> passed      student <=  5000x reference
=> passed      student <=  1000x reference
=> passed      student <=   500x reference
=> BONUS       student <=    10x reference
=> BONUS       student <=     2x reference
=> BONUS       student <=   0.5x reference

Test 4a-c: time length() and ancestor() with random subsets of 100 vertices
  *  digraph-wordnet.txt
     -  reference solution calls per second:   14147.00
     -  student   solution calls per second:  120955.00
     -  reference / student ratio:                 0.12

=> passed      student <= 10000x reference
=> passed      student <=  5000x reference
=> passed      student <=  1000x reference
=> passed      student <=   500x reference
=> BONUS       student <=     2x reference
=> BONUS       student <=   0.5x reference

Test 5: Time 10 calls to length() and ancestor() on random path graphs
        (must handle V = 65536 in under 2 seconds)

             V  seconds
        ---------------
         32768     0.07
         65536     0.18
==> passed


Total: 19/14 tests passed!


================================================================



********************************************************************************
*  TIMING (substituting reference SAP)
********************************************************************************

Timing WordNet
*-----------------------------------------------------------
Running 11 total tests.

Test 1: check that exactly two In object created
        (one for synsets file and one for hypernyms file)
==> passed

Test 2: count number of SAP operations when constructing a WordNet object
        and calling distance() and sap() three times each
  * calls to constructor = 0
    - minimum required   = 1
    - maximum allowed    = 1

  * calls to length()    = 0
    - minimum required   = 3
    - maximum allowed    = 3

  * calls to ancestor()  = 0
    - minimum required   = 3
    - maximum allowed    = 3

==> FAILED

Test 3: count Digraph operations during WordNet constructor
  * synsets = synsets.txt; hypernyms = hypernyms.txt
  * number of synsets    = 82192
  * number of hypernyms  = 84505
  * calls to constructor = 1
  * calls to addEdge()   = 84505
  * calls to adj()       = 0
  * calls to outdegree() = 0
  * calls to indegree()  = 0
  * calls to reverse()   = 0
  * calls to toString()  = 0

==> passed

Test 4: count Digraph operations during 1000 calls each
        to distance() and sap()
  * synsets = synsets.txt; hypernyms = hypernyms.txt
  * calls to constructor = 0
  * calls to addEdge()   = 0
  * calls to adj()       = 124410
    - the WordNet digraph has 82192 vertices
    - average number of student   calls to adj() per query = 124.4
    - average number of reference calls to adj() per query = 45.8

  * calls to reverse()   = 0
  * calls to toString()  = 0

==> FAILED
```

```
Test 5: time WordNet constructor
  * synsets = synsets.txt; hypernyms = hypernyms.txt
    - student constructor time =  0.21 seconds
    - maximum allowed     time = 10.00 seconds

==> passed

Test 6a-e: time sap() and distance() with random nouns


        ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
        OperationCountLimitExceededException
        Number of primitive operations in Digraph exceeds limit: 2000000000
        ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

==> FAILED

Test 7: time isNoun() with random nouns
  * synsets = synsets.txt; hypernyms = hypernyms.txt
    - reference solution calls per second: 1060405.00
    - student   solution calls per second:  863245.00
    - reference / student ratio:                 1.23
    - allowed ratio:                             4.00
==> passed

Total: 4/11 tests passed!


================================================================



********************************************************************************
*  TIMING (substituting reference SAP and WordNet)
********************************************************************************

Timing Outcast
*-----------------------------------------------------------
Running 2 total tests.

Test 1: count calls to methods in WordNet
 * outcast4.txt
    - student   distance() calls  = 0
    - reference distance() calls  = 6
    - maximum allowed             = 16

 * outcast10.txt
    - student   distance() calls  = 0
    - reference distance() calls  = 45
    - maximum allowed             = 100

 * outcast29.txt
    - student   distance() calls  = 0
    - reference distance() calls  = 406
    - maximum allowed             = 841

==> FAILED

Test 2: timing calls to outcast() for various outcast files

Total time must not exceed 1.0 seconds.

    filename       n     time
-----------------------------
   outcast4.txt    4     0.00
   outcast5.txt    5     0.00
  outcast5a.txt    5     0.00
   outcast5.txt    5     0.00
   outcast7.txt    7     0.00
   outcast8.txt    8     0.00
  outcast8a.txt    8     0.00
  outcast8b.txt    8     0.00
  outcast8c.txt    8     0.00
   outcast9.txt    9     0.00
  outcast9a.txt    9     0.00
  outcast10.txt   10     0.00
 outcast10a.txt   10     0.00
  outcast11.txt   11     0.00
  outcast12.txt   12     0.00
 outcast12a.txt   12     0.00
  outcast20.txt   20     0.00
  outcast29.txt   29     0.00

Total elapsed time: 0.00 seconds

==> passed


Total: 1/2 tests passed!


================================================================
```